

对外经济贸易大学

2004 年攻读硕士学位研究生入学考试

《应用软件基础》试题 (试题代码: 441)

注意: 请将所有答案写在答题纸上

C 语言程序设计部分 (共 75 分)

一、选择题 (下列各题 A, B, C, D 四个选项中, 只有一个选项是正确的。请选择正确选项) (每小题 1 分, 共 20 分)

1. 设 $\text{int } i = 6, j$; 则表达式 $j = (i++) + (++i), ++j$ 的值是 【1】

- A) 4
B) 13
C) 14
D) 15

2. 执行以下语句后, a 和 b 的值是 【2】

$\text{int } a = 5, b = 6, w = 1, x = 2, y = 3, z = 4;$
 $(a = w > x) \&\& (b = y < z);$

- A) 5 和 6
B) 0 和 6
C) 0 和 0
D) 0 和 1

3. 下面程序段的运行结果是 【3】

$\text{int } n = 0;$
 $\text{while } (n++ <= 2);$
 $\text{printf} (" \%d", n);$

- A) 4
B) 3
C) 2
D) 有语法错

4. 下列程序段不是死循环的是 【4】

A) $\text{int } i = 100;$
 $\text{while}(1)$
 $\{ i = i \% 100 + 1;$
 $\text{if } (i > 100) \text{ break};$
 $\}$

B) $\text{for } (; ;)$

C) $\text{int } k = 0;$
 $\text{do } \{ ++k; \} \text{ while } (k >= 0);$

D) $\text{int } s = 36;$
 $\text{while } (s);$
 $--s;$

5. 下面程序的运行结构是 【5】:

$\#include <stdio.h>$
 $\text{main}()$
 $\{ \text{char } \text{str}[] = "SSSWLIA", c;$
 $\text{int } k;$
 $\text{for } (k = 2; (c = \text{str}[k]) != '\0'; k++)$
 $\{ \text{switch } (c)$

```

{ case 'I': ++k; break;
  case 'L': continue;
  default: putchar (c); continue;
}
putchar ('*');
}

```

A) SSW*

B) SW*

C) SW*A

D) SW

6. 下列程序中, 有 3 个能够完成求 1 到 100 的累加和的功能, 另一个所完成的功能与其他程序不同, 它是【6】 **B**

(A) `int s=0, i=0;`
`while(i <= 100)`
`s += i++;`

(B) `int s=0, i=0;`
`while(i <= 100)`
`s += ++i;`

(C) `int s=0, i=0;`
`while(i < 100)`
`s += ++i;`

(D) `int s=0, i=0;`
`while(++i <= 100)`
`s += i;`

7. 若有如下定义:

`int x=1, y=0;` 则表达式 `x-->(y+x)?5:25>y++?'1':'2'` 的值是【7】 **A**

(A) 5

(B) 25

(C) '1'

(D) '2'

8. 以下各组选项中, 均能正确定义二维实型数组 a 的是【8】: **C**

A) `float a[3][4];`
`float a[][4];`
`float a[3][]={{1},{0}};`

B) `float a(3,4);`
`float a[3][4];`
`float a[][]={{0},{0}};`

C) `float a[3][4];`
`float a[][4]={{0},{0}};`
`auto float a[][4]={{0},{0},{0}};`

D) `float a[3][4];`
`float a[3][];`
`float a[][4];`

9. 调用 `strlen("abcd\0ef0g")` 的返回值是【9】 **D**

(A) 9

(B) 8

(C) 7

(D) 4

10. 下面函数调用语句含有实参的个数为【10】 **B**

`func((exp1, exp2), (exp3, exp4, exp5))`

(A) 1

(B) 2

(C) 4

(D) 5

11. 下面程序的运行结果是【11】 **B**

```

#include <stdio.h>
main()
{int k=4, m=1, p;
p = fun(k,m); printf("%d", p);
p = fun(k,m); printf("%d", p);
}

```

```

fun( int a, int b )
{static int m = 0, i = 2;

```

```

    if (a < m)
        m = a;
    if (b < i)
        i = b;
    return m + i;
}

```

```

fun( int a, int b )
{static int m = 0, i = 2;

```

```

    return m + i;
}

```

```

i += m + 1;   i = 3   i = 11
m = i + a + b;   m = 3 + 5 = 8   m = 16
return (m);
}

```

A) 8, 15

B) 8, 16

C) 8, 17

D) 8, 20

12. 下面程序的运行结果是【12】

#include <string.h>

main()

{ char a[7] = "abcdef", b[4] = "ABC";

strcpy (a, b);

printf (" %s %c ", a, a[5]);

}

A) ABC

C) ABCe

B) ABC0

D) ABCf

13. 已知: int a[3][4], *p = &a[0][0]; p += 6; 那么 *p 和【13】的值相同

A) a[1][2]

B) *(&a[0]+6)

C) *(a[1]+2)

D) *(a+6)

14. 已知: int *p; 使用如下语句动态申请一个 int 型的存储单元, 应当使用【14】

A) (int *)

B) int *

C) (*int)

D) int

15. 某函数的功能是交换 x 和 y 中的值, 并且通过正确调用返回交换的结果。

能正确执行此功能的函数是【15】

A) funa (int *x, int *y);

B) funa (int *x, int *y);

{ int *p;

{int t;

*p = x; *x = *y; *y = *x;

t = x; x = y; y = t;

}

}

C) func (int *x, int *y);

D) fund (int *x, int *y);

*x = *y; *y = *x;

{*x = *x + *y; *y = *x - *y;

}

*x = *x - *y; }

16. 以下程序的输出结果是【16】

#include <stdio.h>

#include <string.h>

main()

{static char a[] = "ABCDEFGH";

fun(a, strlen(a));

puts(a);

}

void fun(char *w, int m)

{char s, *p1, *p2;

p1 = w;

p2 = w+m-1;

while (p1 < p2)

{ s = *p1++; *p1 = *p2--; *p2 = s; }

- }
 A) ABCDEFG
 C) ABGFCD
 B) AGAAGAG
 D) GFEDCBA
17. 以下程序的输出结果为【17】
- ```

#define PT 5.5
#define S(x) PT*x*x
main()
{ int a=1, b=2;
 printf("%4.1fn", S(a+b));
}

```
- A) 9.5  
 C) 12.5  
 B) 12.0  
 D) 33.5
18. 若有以下程序段:
- ```

struct dent
{ int n;
  int *m;
};
int a=1, b=2, c=3;
struct dent s[3]={ {101, &a}, {102, &b}, {103, &c} };
main()
{ struct dent *p=s;
  ...
}

```
- 则以下表达式中值为2的是【18】
- A) (p++)->m
 C) (*p).m
 B) *(p++)->m
 D) *(++p)->m
19. 下面程序的运行结果是【19】
- ```

fun (int **a, int p[2][3])
{ **a=p[1][1]; }
main()
{ int x[2][3]={ 2, 4, 6, 8, 10, 12 }, *p;
 p=(int *) malloc(sizeof(int));
 fun (&p, x);
 printf("%d\n", *p);
}

```
- A) 6  
 C) 10  
 B) 8  
 D) 12
20. 阅读以下程序段及对程序功能的描述, 其中正确的描述是【20】
- ```

#include <stdio.h>
main()
{ FILE *in, *out;
  char ch, infile[10], outfile[10];
  printf( "Enter the infile name :\n");
  scanf( "%s", infile );
}

```

```

printf("Enter the outfile name :\n");
scanf("%s", outfile);
if((in = fopen(infile, "r")) == NULL);
{ printf("can not open infile\n");
  exit(0);}
if((out = fopen(outfile, "w")) == NULL);
{ printf("can not open outfile\n");
  exit(0);}
while(!feof(in)) fputc(fgetc(in), out);
fclose(in);
fclose(out);
}

```

- A) 程序完成将磁盘文件的信息在屏幕上显示的功能
 B) 程序完成将两个磁盘文件合二为一的功能
 C) 程序完成将一个磁盘文件复制到另一个磁盘文件中
 D) 程序完成将两个磁盘文件合并并且在屏幕上输出

二、阅读程序, 回答问题 (每小题 4 分, 共 16 分)

1. 执行下面程序时, 若输入: 52 (回车)

则下面程序段的输出结果是【1】: 采用此算法的前提条件是数组 a 必须【2】

```

#include<stdio.h>
main()
{ int a[8] = {6, 12, 18, 42, 44, 52, 67, 94};
  int low = 0, mid, high = 7, found, x;
  found = 0;
  scanf("%d", &x);
  while((low <= high) && (found == 0))
  { mid = (low + high) / 2;
    if(x > a[mid]) low = mid + 1;
    else if(x < a[mid]) high = mid - 1;
    else {found = 1; break;}
  }
  if(found == 1)
    printf("Search Successful! The index is :%d\n", mid);
  else printf("Can not search!\n");
}

```

search successful! The index is 5

4 号排序

2. 下面程序段的输出结果是【3】: 功能是【4】

```

main()
{long k;
  k = fun(7);
  printf("k = %d\n", k);
}

```

k=13

输出斐波那契数列的第 7 项

```

long fun(int g)
{ switch (g)
  { case 0: return 0;
    case 1: case 2: return 1;
  }
  return (fun(g-1) + fun(g-2));
}

```

$$\begin{aligned}
 f(7) &= f(6) + f(5) \\
 f(6) &= f(5) + f(4) & f(5) &= f(4) + f(3) \\
 f(4) &= f(3) + f(2) & f(3) &= f(2) + f(1) \\
 5 & & 3 & & 3 & & 2
 \end{aligned}$$

三、程序选择填空(每空 4 分,共 12 分)

以下程序的功能是: 读入一行字符 (如: a, ..., y, z), 按输入时的逆序建立一个链接式的结点序列, 即先输入的位于链表尾 (形成 z, y, ..., a, 即 z 在链表的首部, a 在链表的尾部), 然后再按输入的相反顺序输出 (输出顺序为 z, y, ..., a), 并释放全部结点。请选择填空。

```

#include "stdio.h"
#define getnode( type ) 【1】 malloc ( sizeof ( type ))
main()
{ struct node
  { char info;
    struct node *link;
  } *top, *p;
  char c;
  while ((c = getchar()) 【2】 B
  { p = getnode(struct node);
    p->info = c;
    p->link = top;
    top = p;
  }
  while (top)
  { 【3】 : AB top = top->link;
    putchar (p->info);
    free(p); }
}

```

- | | |
|----------------|-------------|
| 【1】 A) (type) | B) (type *) |
| C) type | D) type * |
| 【2】 A) == '\0' | B) != '\0' |
| C) == '\n' | D) != '\n' |
| 【3】 A) top = p | B) p = top |
| C) p = top | D) top = p |

四、程序填空(每空 3 分,共 15 分)

程序功能: 从键盘上顺序输入整数, 直到输入的整数小于 0 时才停止输入, 然后反序输出这些整数。请填写完成程序。

```
#include <stdio.h>
```

1 1 2 3 5 8 13

```

stru data
{ int x;
  struct data *link;
} *p;
input()
{ int num;
  struct data *q; Do
  { printf("Enter data:");
    scanf("%d", &num);
    if (num < 0)
      [1]: return; // return;
    q = [2] p; // return;
    q->next = num;
    q->link = p;
    p = q;
    [3]: scanf("%d", &num); } while(1); // while(1);
  }
main()
{ printf("Enter data until data<0:\n");
  p=NULL;
  input();
  printf("Output:");
  while ([4]) p != NULL p != NULL p != NULL
  { printf("%d\n", p->x);
    [5] p = p->link;
  }
}

```

五、编写程序(共 12 分):

编写函数 void sort_out(int *p, int n), 函数的功能是将主函数 a 数组中的整数按升序排序并输出。要求奇数在前, 偶数在后。比如, 如果 a 数组中有 10 个数, 他们是 10 8 9 6 7 5 1 4 3 2, 则输出为 1 3 5 7 9 2 4 6 8 10。主函数如下所示。

```

main()
{ int a[10], n;
  void sort_out(int *p, int n);
  ...
  ...
  n = 10;
  sort_out(a, n);
  for(n = 0; n < 10; n++) printf("%4d", a[n]);
  ...
  ...
}

```

数据库系统概论部分 (共 75 分)

一、选择题(下列各题 A,B,C,D 四个选项中,只有一个选项是正确的,请选择正确选项)(每小题 1 分,共 10 分)

- 数据库系统的核心是【1】。 **B**
 A) 数据库 B) 数据模型
 C) 数据库管理系统 D) 操作系统
- 数据库中产生数据不一致的根本原因是【2】。 **A**
 A) 并发操作 B) 数据冗余
 C) 数据太多 D) 没有约束条件
- 数据模型的组成要素中,【3】描述了系统的静态特性。
 A) 数据结构 **C** B) 数据操作
 C) 数据模型 D) 约束条件
- 下列【4】是关系模型的特点之一。
 A) 结构简单 **A** B) 存取路径不透明
 C) 实体与实体间联系都用关系表示 D) 数据查询效率高
- 下列【5】是 SQL 语言不具备的功能。
 A) 数据定义 **D** B) 数据操纵
 C) 数据控制 D) 关系规范化
- 在关系数据库中,任何二元关系模式的最高范式是【6】。 **A**
 A) BCNF B) 3NF
 C) 2NF D) 1NF
- 在数据库设计各阶段中,概念设计是整个数据库设计的关键,概念设计的结果是产生数据库的【7】。 **B**
 A) 数据流图 B) E-R 图
 C) 模式 D) 外模式
- 数据库转储时转储全部数据库称【8】。 **C**
 A) 动态转储 B) 静态转储
 C) 海量转储 D) 增量转储
- 数据库安全性控制的最外层安全保护措施是【9】。 **D**
 A) 存取控制 B) 数据加密
 C) 视图机制 D) 用户标识与鉴别
- 若有关系模式 $R(U, F)$, $U=\{A, B, C, D, E\}$, $F=\{A \rightarrow D, E \rightarrow D, D \rightarrow B, BC \rightarrow D, DC \rightarrow A\}$, 则【10】是候选码。
 A) AE **A** B) DE
 C) BE D) CE

二、判断题(认为表述正确,填写“T”,否则填写“F”。每小题 1 分,共 5 分)

- (**F**) 关系数据库中“关系模式”是型,“关系”是值。
- () 关系代数表达式可以等价转换为关系演算表达式。
- (**/**) 关系中主属性的值能唯一确定一个元组。
- () 函数依赖集的最小函数依赖集不唯一。
- (**F**) “事务”和“程序”一样是一个不可分的逻辑工作单位。

三、简答题(每小题 5 分,共 20 分)

关系模式是对关系的描述,一般表示为 关系名(属性1,属性2,...属性n)。

关系就是指通常说的一张二维表

1. 简述数据库中“关系”、“关系模型”、“关系模式”、“关系数据库模式”的概念。
关系模型是一种数据模型;在关系模型中,数据的逻辑结构是一张二维表,由行和列组成。
在关系模型中,实体及实体间的联系都是用关系表示。
2. 简述数据库系统的三级模式结构和二级映射。

3. 简述查询优化在关系数据库系统中的必要性及其一般准则。
存取路径对用户透明,查询效率高;准则:①选择运算应尽可能先做(能使计算中中间结果变小)
②执行有连接的关系处理。方法一:在直接属性上建立索引(索引连接)
方法二:对关系排序(排序合并连接方法)
4. 简述数据库中封锁机制可能带来的问题和解决方法。

四、设计题(10分)

某酒店用计算机来管理经营,要建立数据库,需要管理的信息如下:

1. 客房信息:每间客房的客房号和类型号;
 2. 房价信息:每类客房的类型号和类型名及其价格;
 3. 酒店客户的个人信息:客户代号,身份证号,姓名,性别,年龄,工作单位,所在城市。
 4. 酒店客户的入住信息:酒店客户每次入的客房号,入住日期,退房日期。
- 根据上述信息,画出 E-R 图,并按关系模型的要求写出关系模式,标出每个关系模式的主码。

五、SQL 语言(每小题 5 分,共 30 分)

某市商业连锁店管理数据库有若干个表,

1. 分店(分店编号,店名,店址,经营面积,经理姓名)
 2. 商品(商品编号,商品名称,库存量)
 3. 供应商(供应商编号,供应商名称,城市)
 4. 订购(订单号,商品编号,商品数量,应付金额,到货日期,供应商编号)
 5. 配送(商品编号,分店编号,配送商品数量,送货日期,送货员)
 6. 销售(商品编号,分店编号,销售商品数量,销售金额,销售日期)
- (注:“销售商品数量”是一天的合计,带双下划线的字段是表的主键)

按下列查询要求写出 SQL 语句:

1. 查询 2002 年 10 月 10 日 配送给店名为“AAAA”的分点的商品名称和配送商品数量。

2. 统计 2002 年 1 月各分店的销售额,并按从大到小顺序输出。

3. 查询“天津”供应商在“2003 年 9 月 8 日”到货的“方便面”的数量。

4. 统计 2003 年上半年销售“电视机”的数量和金额。

5. 统计经营面积在 1 万平方米以上的分店的个数和它们 2002 年全年的销售金额合计。

6. 查询 2002 年一季度到货的供应商名为“GGGG”应付金额合计是多少。

3. SELECT COUNT(订购.商品数量)

FROM 订购,供应商,商品

WHERE 订购.供应商编号=供应商.供应商编号 AND

订购.商品编号=商品.商品编号 AND

商品.商品名称='方便面' AND 供应商.城市='天津'

4. SELECT COUNT(销售商品数量), COUNT(销售金额)

FROM 销售,商品

WHERE 销售.商品编号=商品.商品编号 AND

商品.商品名称='电视机' AND

销售.销售日期 BETWEEN '2002-01-01' AND '2002-03-31'